

CPS - Wareneingang

1. Kurze Produktbeschreibung

In einer perfekten Welt müssen Menschen keinen monotonen Aufgaben nachgehen, wie zum Beispiel dem Sortieren von Waren. Es würde Systeme geben, die dies vollständig automatisieren und uns Menschen mehr Zeit für andere Tätigkeiten geben würden.

Unser Projekt vereint Sensorik mit einem Roboterarm, um den Wareneingangsprozess in einer Fabrik zu vereinfachen und im besten Fall sogar zu automatisieren. Die Sensoren erfassen kontinuierlich Daten über die ankommenden Waren. Diese erfassten Daten werden umgehend analysiert, um den geeignetsten Ablageort zu bestimmen. Der Roboterarm, welcher über WIFI mit eingebauten Mikrocontrollern verbunden ist, übernimmt die Aufgabe des Sortierens in geeignete Container.

Wenn die Budgets es zulassen, wird es eine GUI geben, über die man von jedem Ort zu jeder Zeit den aktuellen Stand des Wareneingangs beobachten kann.

Dieses System verringert die Fehleranfälligkeit beim Sortieren der Waren und erhöht darüber hinaus die Effizienz des gesamten Lagerbetriebes, da Mitarbeiter sich anderen Aufgaben widmen können.

Unsere Lösung ist für alle Unternehmen geeignet, die sich weniger monotone Sortierarbeit oder einen effizienteren Lagerbetrieb wünschen.

2. Zielbestimmungen

Es wird ein Wilkata Mirobot Roboterarm zur automatischen Warensortierung genutzt. Um diesen zu steuern sind folgende Funktionen vorgesehen:



2.1 Bedienung

Zur Bedienung des Systems sind für den Nutzer folgende Möglichkeiten vorgesehen:

Einschaltknopf

Zum An- und Ausschalten des Roboterarms ist ein Schalter vorgesehen.

Reset-Button

Wenn eine Ladung sortiert ist, kann mit dem Reset-Button das System zurückgesetzt werden, damit eine neue Ladung sortiert werden kann.



2.2 Sensoren

Um diese Sortierung zu automatisieren, werden Sensoren gebraucht, die die Messwerte feststellen, aufgrund derer der Roboterarm gesteuert werden kann. Folgende Sensoren sind bis jetzt vorgesehen:

Waage

Mit einer Waage wird zu sortierende Ware gewogen.

Lichtschranken

Um festzustellen, ob die Kisten (2) voll sind, werden Lichtschranken eingesetzt.



3. Architektur und Techstack

Die Architektur des Systems basiert auf einer Kombination von Raspberry Pi und dem Arduino Microcontroller, die miteinander über WLAN kommunizieren. Diese beiden Komponenten übernehmen jeweils spezifische Aufgaben, um den automatisierten Wareneingang durch einen Wlkata Mirobot zu realisieren.

3.1 Arduino Uno Wifi

Der Arduino fungiert als zentrale Steuerungseinheit für alle weiteren Funktionalitäten unseres Produkts. Dazu werden die nötigen Sensoren analog mit dem Arduino verbunden und die berechneten Bewegungen für eine Sortierung des Objekts mittels eines WIFI-Moduls drahtlos an einen Raspberry Pi übertragen. Die Programmierung des Mikrocontrollers wird in C und C++ erfolgen, gängige Bibliotheken und Funktionen des Arduino kommen hierfür zum Einsatz.

3.2 Raspberry Pi (Gen. 5)

Der Raspberry Pi übernimmt ausschließlich die Steuerung des Wlkata Mirobot Roboterarms, alle Befehle erhält er direkt vom Arduino und setzt diese um, um schließlich die Ware in die entsprechenden Container zu sortieren. Für die **Implementation** nutzen wir Python.

4. Qualitätssicherung

Um die Qualität der Entwicklung zu sichern, finden parallel zur Entwicklung diverse Tests und Debuggingmaßnahmen statt. Essentiell hierbei ist der bewährte GitFlow-Workflow, der es den einzelnen Teammitgliedern individuell ermöglicht, Features zu entwickeln und isoliert zu testen, bevor die **funktionale und getestete Funktionalitätseinheit** ins letztendliche Produkt einfließt. Ferner wird auch zwischen den Entwickler:innen auf ein hohes Maß an Kommunikation z.B. durch den Einsatz eines Discord-Servers gesetzt, um die Interoperabilität zwischen den verschiedenen Layern der Architektur noch innerhalb der Implementationsphase sicherzustellen.

In der Phase bis zum letzten Meilenstein (siehe unten) startet die formale Testphase, in der insbesondere mit Belastungstests sichergestellt wird, dass das Produkt auch unter kritischen Bedingungen seine Funktionalität wahren kann.

Folgende Metriken gelten für den Steuerungscode des Roboters, aber auch für den Code, der auf dem Arduino laufen soll:

1. Method Lines of Code: maximal 30
2. Zyklomatische Komplexität nach McCabe: maximal 10
3. Nested Block Depth: maximal 5
4. Anzahl Parameter pro Funktion: maximal 10



5. Organisation

Die Entwickler sind:

- Erik Haarländer
- Eleni Kiometzis
- Sebastian Schmidt
- Anton Schubert
- Vladislav Schulmeister

5.1 Rollenverteilung

Projektleiter: Anton Schubert

Dokumentationsbeauftragte: Eleni Kiometzis

Qualitätssicherungsbeauftragter: Erik Haarländer

Die Aufgabenzuweisung für die Planung und das Erreichen der Meilensteine erfolgt durch die Projektleitung bzw. durch einen ernannten Themenverantwortlichen direkt.

5.2 Aufgabenbestimmung

In einem wöchentlichen Meeting plant das Team zusammen ihre jeweils einwöchigen Sprints. Dabei wird besprochen, welche Aufgaben in der kommenden Woche angegangen werden und wie diese verteilt werden. Außerdem gibt es ein Update über den Fortschritt des Gesamtprojektes. Zusätzlich werden die Aufgaben für einen besseren Überblick in einem Kanban Board mit den Spalten: Open, Todo, Doing und Closed eingetragen.



Für die allgemeine Kommunikation wird ein eigens erstellter Discord-Server benutzt und für die Version Control wird der Gitlab-Server der Technischen Universität Berlin verwendet.

6. Entwicklungsplan

Das Projekt ist in drei Phasen aufgeteilt, die das Äquivalent zu den Meilensteinen sind.

Zum Ende jeder Phase wird eine Präsentation stattfinden, um die Auftraggeber über den Stand des Projektes zu informieren.

6.1 Phase 1:

(30.05. - 15.05.2014)

Das Ziel der ersten Phase ist es, einen ersten Prototyp zu entwickeln, auf den man in späteren Phasen aufbauen kann. Außerdem soll dieser Prototyp als *proof of concept* dienen.

- Kommunikation zwischen Raspberry-Pi und Arduino
- Auslesen der Sensoren
- Greifen und Ablegen der Pakete mit Hilfe des Roboters (Abhängig von der Verfügbarkeit des Arms)
- Planung der verschiedenen Funktionen per UML

6.2 Phase 2:

(20.05. - 12.06.2024)



Das Ziel der zweiten Phase ist es, aufbauend auf den Prototyp aus Phase 1, 60% der Funktionen zu implementieren.

- Sortierung nach Gewicht (Software)
- Vereinigung der Komponenten der ersten Phase
- Ausführung spezifischer Demoläufe

6.3 Phase 3:

(17.05. - 05.07.2024)

Das Ziel der dritten Phase ist es, die restlichen Funktionalitäten zu implementieren und das Produkt abgabebereit zu machen. Nach Möglichkeit werden noch die optionalen Features, die im siebten Punkt definiert sind, implementiert.

In Phase 2 und 3 werden die **implantierten** Funktionen umfassend getestet.

7. Optionale Features

Abhängig von der Zeit ist geplant, folgende Funktionen zusätzlich zu implementieren:

1. GUI - Eine grafische Oberfläche auf einem zusätzlichen Raspberry Pi zur angenehmen Steuerung des Systems und eventuellen Extrafunktionen
2. Individualisierung des Sortierungsprozesses auf der GUI durch
 - a. zusätzliche Kisten
 - b. Auswahl des Gewichtes nach dem sortiert werden soll
 - c. Auswahl des gewünschten Füllstands der Kisten
 - d. Pausierung des Prozesses